

Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática



**Estudo comparativo entre ferramentas de
Gerência de Requisitos**

Trabalho de Graduação

Aluno: Rafael Richa Teixeira Ananias

Orientador: Prof. Dr. Alexandre Marcos Lins de Vasconcelos

Recife/2009

Rafael Richa Teixeira Ananias

Estudo comparativo entre ferramentas de
Gerência de Requisitos

Trabalho apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, pelo Centro de Informática da Universidade Federal de Pernambuco, sob orientação do Prof. Dr. Alexandre Marcos Lins de Vasconcelos.

Recife/2009

Agradecimentos

Primeiramente agradeço a Deus por me proporcionar força e perseverança, não só neste trabalho, mas em toda a vida, mantendo-me firme diante dos obstáculos.

Agradeço à minha família, meu pai Nilson Ananias, minha mãe Glaucia Ananias, meu irmão Cesar Ananias, minha noiva Cristeane Sampaio, meu avô Widomar Teixeira, minhas avós Gloria Richa e Nilsa Ananias e a todos outros por todo amor, compreensão, apoio, carinho e dedicação dados a mim. Sem eles nada disso seria possível. Meu amor é de vocês.

Ao meu orientador, Professor Doutor Alexandre Marcos Lins de Vasconcelos, por todo apoio, paciência, prontidão e dedicação dados a mim no decorrer deste trabalho. Sua ajuda foi fundamental.

Agradeço a todos meus amigos que contribuíram de alguma forma para o sucesso deste trabalho e do curso.

Resumo

Este trabalho tem como finalidade fazer um estudo comparativo entre ferramentas gratuitas de Gerência de Requisitos, fortes aliadas no processo de desenvolvimento de sistemas. Pretende servir como apoio na escolha do programa mais apropriado ao ambiente dos interessados. Neste trabalho, são utilizadas três normas internacionais, que propõem um sistema de avaliação para que seja observada a qualidade dos produtos de software. As ferramentas selecionadas a serem submetidas ao estudo são: Jeremia, OSRMT, Tiger Pro e Xuse.

Palavras-chave: Gerência de Requisitos, ferramentas de Gerência de Requisitos.

Sumário

Agradecimentos	3
Resumo	4
Índice de Figuras	8
Índice de Tabelas	9
1 Introdução	10
1.1 Objetivos	11
1.2 Estrutura do trabalho	11
2 Gerência de Requisitos	12
2.1 Processo de Engenharia de Requisitos	12
2.2 Gerência de Requisitos	14
2.2.1 Gerenciamento de mudanças	15
2.3 Considerações finais	17
3 Gerência de Requisitos em modelos de qualidade	18
3.1 CMMI	18
3.1.1 SG1 – Gerenciar Requisitos	21
3.2 MPS-BR	22
3.3 Considerações finais	26
4 Modelos de qualidade de produto de software	27
4.1 ISO/IEC 9126	27
4.1.1 ISO/IEC 9126-1	28
4.1.2 ISO/IEC 9126-2	31
4.1.3 ISO/IEC 9126-3	31
4.1.4 ISO/IEC 9126-4	32

	6
4.2 ISO/IEC 14598	33
4.3 ISO/IEC 12119	36
4.3.1 Requisitos de Qualidade.....	36
4.3.2 Instruções para Testes	37
4.4 Considerações finais	38
5 Processo de avaliação	39
5.1 Usuários	40
5.2 Características, sub-características e atributos.....	40
5.3 Distribuição dos pontos	43
5.4 Sistema de métricas adotado.....	45
5.4.1 Cálculo da nota final	45
5.5 Considerações finais	46
6 Análise das ferramentas selecionadas.....	47
6.1 Jeremia	47
6.2 OSRMT.....	48
6.3 Tiger Pro	49
6.4 Xuse	50
6.5 Análise das ferramentas	51
6.5.1 Jeremia	52
6.1.2 OSRMT	53
6.1.3 Tiger Pro.....	54
6.1.4 Xuse.....	55
6.2 Comparação entre as ferramentas	56
6.3 Análise dos resultados	56

	7
6.3.1 Jeremia	56
6.3.2 OSRMT	57
6.3.3 TigerPro.....	58
6.3.4 Xuse.....	59
6.4 Considerações finais	60
7 Conclusão	61
7.1 Dificuldades encontradas	62
7.2 Trabalhos futuros	62
8 Glossário.....	63
9 Referências bibliográficas	64

Índice de Figuras

Figura 1. Modelo de processo de Engenharia de Requisitos de alto nível	13
Figura 2. Processo de Gerenciamento de Mudanças	15
Figura 3. Tipos de Rastreamento	16
Figura 4. Representação contínua do CMMI.....	19
Figura 5. Representação por estágios do CMMI.....	20
Figura 6. Níveis do MPS-BR.....	23
Figura 7. ISO/IEC 9126-1	30
Figura 8. ISO/IEC 14598-1	33
Figura 9. Estrutura da ISO/IEC 12119	36
Figura 10. Distribuição dos atributos e sub-características de uma característica	39
Figura 11. Tela inicial do Jeremia	48
Figura 12. Tela inicial do OSRMT.....	49
Figura 13. Tela inicial do Tiger Pro	50
Figura 14. Tela inicial do Xuse	51
Figura 15. Funcionamento do TigerPro.....	59

Índice de Tabelas

Tabela 1. Distribuição dos pontos	44
Tabela 2. Tabela de análise do Jeremia	52
Tabela 3. Tabela de análise do OSRMT.....	53
Tabela 4. Tabela de análise do Tiger Pro	54
Tabela 5. Tabela de análise do Xuse	55
Tabela 6. Tabela de comparação das ferramentas	56

1 Introdução

“Aos requisitos estão associados os principais problemas do desenvolvimento de software. Requisitos que não refletem as reais necessidades dos usuários, incompletos e/ou inconsistentes, mudanças em requisitos já acordados e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores são as principais dificuldades relatadas, provocando re-trabalho, atrasos no cronograma, custos ultrapassados e a insatisfação dos clientes e usuários de software” (BLASCHEK, 2002, p.1).

Os requisitos descrevem basicamente as características e propriedades de um sistema, o que o sistema deve ou não fazer, bem como suas restrições. Elicitação, documentação, análise e validação de requisitos fazem parte de um conjunto de atividades executadas pelos desenvolvedores, chamado de Processo de Engenharia de Requisitos. Uma organização que deseja desenvolver sistemas de qualidade, respeitando prazos e custos, deve ter um processo de requisitos bem definido, entendido e utilizado pelos colaboradores.

“Com o passar do tempo, mudanças ocorrem nos requisitos devido a diversos fatores como erros, inconsistências, problemas organizacionais, evolução do conhecimento dos stakeholders, alterações legais, etc., exigindo um grande esforço das empresas para o controle e gerenciamento dos mesmos” (GRANDE, s.d., p.1).

Visando apoiar os responsáveis pelo desenvolvimento de software, foram criadas as ferramentas para gerência de requisitos. Essas ferramentas basicamente têm a capacidade de coletar, armazenar, manter e gerenciar mudanças dos requisitos. Dentre as várias soluções, encontram-se tanto programas pagos quanto gratuitos, cada uma com suas características, que podem, de acordo com a realidade da empresa, se adaptar de forma mais amigável às necessidades.

1.1 Objetivos

O objetivo deste Trabalho de Graduação é fazer um estudo comparativo entre ferramentas de Gerência de Requisitos, podendo servir de apoio aos interessados na escolha da ferramenta apropriada, de acordo com suas necessidades. Esta comparação será feita com a observação e avaliação de critérios baseados em normas internacionais, visando de medir a qualidade dos softwares.

1.2 Estrutura do trabalho

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta a Gerência de Requisitos, onde discute conceitos do Processo de Engenharia de Requisitos. O capítulo 3 aborda a Gerência de Requisitos nos modelos de qualidade, mostrando o CMMI e o MPS-BR. No capítulo 4 são discutidos os modelos de qualidade de produto de software, expondo a ISO/IEC 9126, a ISO/IEC 14598 e a ISO/IEC 12119. O capítulo 5 demonstra o processo de avaliação das ferramentas selecionadas. No capítulo 6 são apresentados os resultados do estudo dos softwares. Finalizando, o capítulo 7 apresenta as conclusões a respeito do trabalho realizado, dificuldades encontradas e trabalhos futuros.

2 Gerência de Requisitos

Gerência de Requisitos é uma atividade que corre em paralelo com os processos da Engenharia de Requisitos. Assim, primeiramente será abordado o processo de Engenharia de Requisitos para posteriormente expor a Gerência de Requisitos propriamente dita.

A Engenharia de Requisitos é um grande processo que envolve todas as atividades que colaboram na produção de um Documento de Requisitos¹ além de sua manutenção com o decorrer do tempo. Este processo deve ser precedido de um estudo de viabilidade, que, de acordo com as restrições do projeto em questão, determinará se é viável ou não dar continuidade ao projeto. O estudo de viabilidade envolve questões como:

- Viabilidade operacional: onde se mede o quão adequado o sistema é para a organização;
- Viabilidade técnica: analisa-se se os recursos técnicos estão disponíveis e se é viável tecnicamente seguir com o projeto;
- Viabilidade de cronograma: estuda-se a razoabilidade do cronograma do projeto;
- Viabilidade econômica: onde se analisa o custo-benefício do projeto.

Após esse estudo, com a conclusão de que o projeto é viável, começa-se o processo de Engenharia de Requisitos.

2.1 Processo de Engenharia de Requisitos

A Engenharia de Requisitos tem como objetivo melhorar a modelagem de sistemas e a capacidade de analisá-los, permitindo melhor entendimento de suas características antes da implementação.

¹ Documento que contém os serviços e funcionalidades que o sistema deve prover, restrições, informações sobre o domínio da aplicação, bem como restrições no processo usado para desenvolver o sistema.

Vale ressaltar que não existe um processo único e ideal de gerência de requisitos. O processo varia de acordo com vários fatores de uma organização: maturidade técnica, envolvimento em disciplinas, cultura organizacional, domínio da aplicação dentre outros.

Basicamente esses processos têm os mesmos pilares: elicitación, análise e negociação, documentação e validação de requisitos. A Figura 1 apresenta um exemplo de modelo de processo de engenharia de requisitos de alto nível (KOTONOYA; SOMMERVILLE, 1998, p.32):

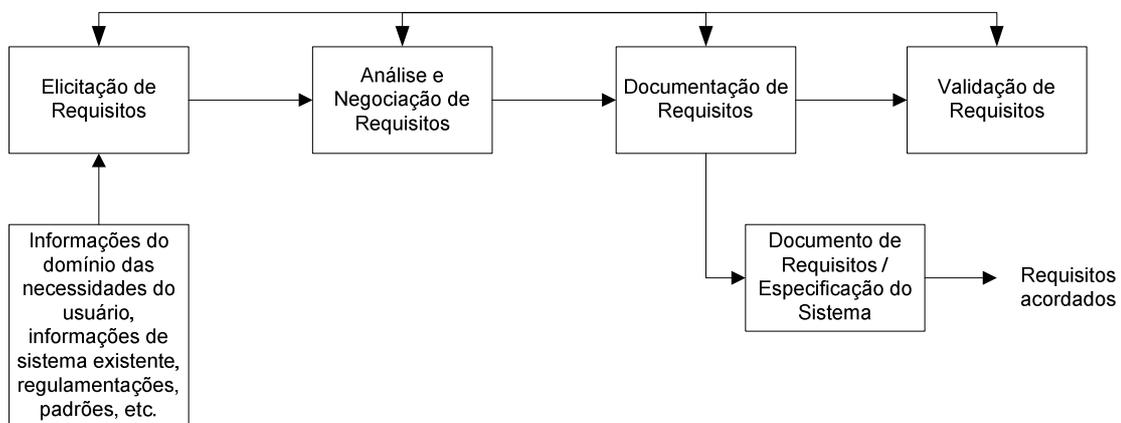


Figura 1. Modelo de processo de Engenharia de Requisitos de alto nível

A **Elicitação de Requisitos** é uma atividade onde os requisitos são descobertos consultando stakeholders², muitas vezes chamada de aquisição de requisitos ou descoberta de requisitos.

Na **Análise e Negociação dos Requisitos** os requisitos são analisados detalhadamente e diferentes stakeholders negociam para definir quais requisitos serão aceitos. Essa atividade é necessária, pois inevitavelmente ocorrem conflitos entre requisitos de diferentes fontes, requisitos incompletos e incompatibilidade de requisitos com o escopo do projeto.

A **Documentação de Requisitos** é a atividade onde os requisitos acordados são documentados de forma apropriada.

² Todas as pessoas envolvidas direta e indiretamente com o projeto.

Na **Validação de Requisitos** os problemas do documento de requisitos são detectados, antes que sejam utilizados no desenvolvimento do sistema.

Agora, a Gerência de Requisitos, que como foi dito caminha em paralelo com os processos de Engenharia de Requisitos, será abordada.

2.2 Gerência de Requisitos

Em todos os estágios do desenvolvimento de sistemas, novos requisitos surgem e mudanças podem ocorrer nos que já existem. Segundo Kotonya e Sommerville (1998, p.113), geralmente essas mudanças ocorrem em 50% dos requisitos antes de serem colocados em atividade. Com isso, fica claro que essas mudanças podem provocar sérios problemas aos desenvolvedores.

Para minimizar os problemas que podem ocorrer, a Gerência de Requisitos se faz necessária. Basicamente, é um processo que permite compreender e controlar mudanças nos requisitos de sistemas. Um estudo europeu revela que o gerenciamento dos requisitos dos clientes é um dos principais problemas no desenvolvimento e produção de sistemas (KOTONOYA; SOMMERVILLE, 1998, p.114).

Os principais objetivos do gerenciamento de requisitos são (KOTONOYA; SOMMERVILLE, 1998, p.114):

- Gerenciar mudanças para requisitos acordados;
- Gerenciar o relacionamento entre requisitos;
- Gerenciar as dependências entre documentos de requisitos e outros documentos produzidos no processo de Engenharia de Software.

2.2.1 Gerenciamento de mudanças

O gerenciamento de mudanças está relacionado a procedimentos, processos e padrões que são usados para gerenciar mudanças nos requisitos do sistema (KOTONOYA; SOMMERVILLE, 1998, p. 116). Essas mudanças podem ocorrer por diversos fatores. Alguns deles são:

- Erros, conflitos e inconsistência dos requisitos;
- Envolvimento do cliente;
- Problemas relacionados à técnica, cronograma e gastos;
- Mudança de prioridade dos clientes;
- Mudanças no ambiente onde funcionará o sistema;
- Mudanças organizacionais.

O processo de gerência de mudanças consiste em uma série de atividades para documentar, reportar, analisar, definir custos e implementar mudanças de um conjunto de requisitos. A Figura 2 representa o processo (KOTONOYA; SOMMERVILLE, 1998, p. 124).

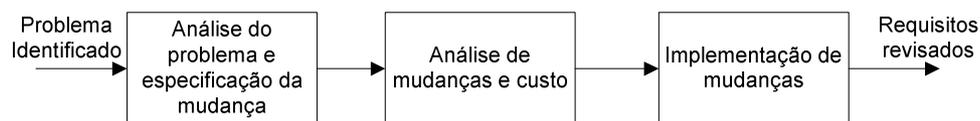


Figura 2. Processo de Gerenciamento de Mudanças

Os três estágios da figura representam:

- Identifica-se algum problema de requisitos, que pode vir da análise de requisitos, novas necessidades dos clientes ou de problemas operacionais do sistema. Com isso, os requisitos são analisados utilizando informações do problema e as mudanças são propostas;

- Analisam-se as mudanças propostas e se toma conhecimento da quantidade de requisitos que serão afetados. Assim, pode-se saber quanto às mudanças custarão (em termos de tempo e gastos financeiros);
- As mudanças são implementadas. Uma série de alterações ou uma nova versão do documento de requisitos é produzida.

Uma parte crítica do processo de gerência de mudanças é a análise de impacto que as mudanças causam no resto do sistema. Para lidar com esse problema, tem-se o conceito de rastreabilidade, que une informações justamente para poder haver uma análise deste impacto. Davis (1993) definiu quatro tipos de rastreamento (KOTONOYA; SOMMERVILLE, 1998, p. 128):

- Rastreamento Backward-from: relaciona requisitos a suas fontes em outros documentos ou pessoas;
- Rastreamento Forward-from: relaciona requisitos ao projeto e componentes de implementação;
- Rastreamento Backward-to: relaciona o projeto e componentes de implementação aos requisitos.
- Rastreamento Forward-to: relaciona outros documentos (que possa ter precedido os documentos de requisito) aos requisitos relevantes.

A Figura 3 resume os tipos citados acima (KOTONOYA; SOMMERVILLE, 1998, p. 129):

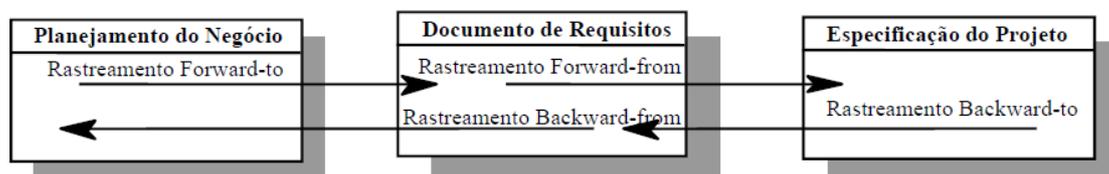


Figura 3. Tipos de Rastreamento

2.3 Considerações finais

Este capítulo abordou a Gerência de Requisitos, atividade que ocorre em paralelo com os processos de Engenharia de Requisitos, que envolvem desde a elicitação até a validação dos requisitos. Também foi possível visualizar o funcionamento do gerenciamento das mudanças bem como a questão da rastreabilidade.

O próximo capítulo abordará os modelos de qualidade CMMI e MPS-BR, mais especificamente o nível de maturidade 2 – Gerenciado e o nível G – Parcialmente Gerenciado, respectivamente. Neles, veremos a relação com a Gerência de Requisitos.

3 Gerência de Requisitos em modelos de qualidade

Neste capítulo, serão abordados dois modelos de qualidade que envolvem a Gerência de Requisitos. Primeiramente será mostrado o CMMI, desenvolvido pela SEI (Software Engineering Institute), em particular o nível de maturidade 2 - Gerenciado. Em seguida será exposto o MPS-BR (Melhoria de Processo do Software Brasileiro), coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), mais especificamente o nível G – Parcialmente Gerenciado.

3.1 CMMI

O CMMI (Capability Maturity Model Integration), modelo reconhecido mundialmente por atestar a maturidade dos processos das organizações, em seu nível de maturidade dois – Gerenciado, possui uma área de processo chamada de Gestão de Requisitos, que visa satisfazer as necessidades das organizações em relação à forma de organizar os requisitos providos pelos clientes, de acordo com o progresso dos projetos. Com isso, pode-se entender claramente o que deve ser feito e o que se espera obter como resultado.

O modelo possui duas representações: contínua e por estágios. Com isso as organizações podem optar por diferentes caminhos para alcançar sua maturidade. Em ambos os casos, o nível dois é “Gerenciado”.

Na representação contínua, têm-se níveis de capacidade e o agrupamento das áreas de processo se dá por categorias. Visa avaliar a capacidade das áreas de processo. A Figura 4 demonstra essa representação:

5 – Otimizado
4 – Gerenciado Quantitativamente
3 – Definido
2 – Gerenciado
1 - Realizado
0 - Incompleto

Figura 4. Representação contínua do CMMI

A seguir, as definições segundo o Modelo de Qualidade CMMI (PENTEADO, 2007, p. 4):

- Nível 0 – Incompleto: um processo é parcialmente realizado ou não realizado;
- Nível 1 – Realizado: um processo realizado satisfaz todos os objetivos específicos da área de processo e produz algum trabalho;
- Nível 2 – Gerenciado: é um processo realizado e também é planejado e executado com políticas pré-definidas;
- Nível 3 – Definido: é um processo gerenciado e ajustado para um conjunto padrão de processos da organização de acordo com suas políticas de conduta;
- Nível 4 – Gerenciado Quantitativamente: um processo definido e controlado com ajuda de técnicas quantitativas e estatísticas;
- Nível 5 – Otimizado: um processo gerenciado quantitativamente, alterado e adaptado para atender aos objetivos de negócio atuais e projetados.

Já a representação por estágios possui níveis de maturidade e o agrupamento das áreas de processo se dá por níveis. Visa avaliar a organização como um todo. A Figura 5 mostra essa representação:

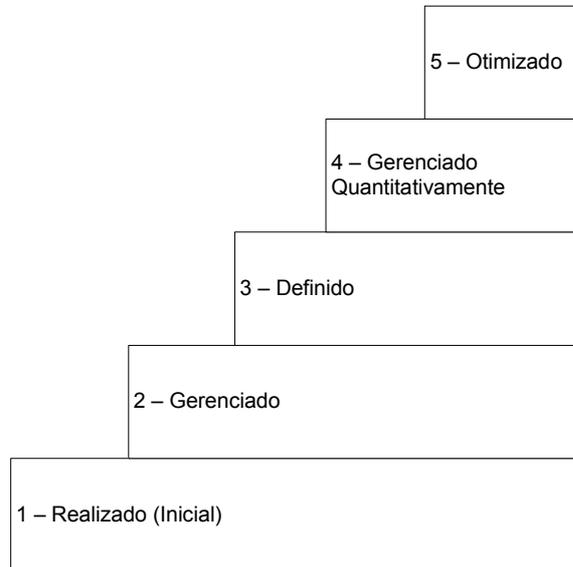


Figura 5. Representação por estágios do CMMI

É possível observar que a representação por estágios, diferentemente da representação contínua não possui o nível 0 – Incompleto.

Os requisitos gerados, incluídos ou impostos pelo projeto devem ser gerenciados de forma que sejam todos tratados. Assim, mudanças serão sempre atualizadas, permitindo o rastreamento desde a fase de elicitação até o produto final.

Esta área de processo (Gestão de Requisitos) possui um único objetivo específico, descrito a seguir:

3.1.1 SG1 – Gerenciar Requisitos

Este SG (Specific Goal) tem como objetivo gerenciar os requisitos e identificar inconsistências nos planos de projeto e quais produtos de trabalho tratados. Possui cinco práticas específicas:

➤ SP1.1 – Obter um entendimento dos requisitos

Esta SP (Specific Practice) se ocupa em saber os significados reais dos requisitos elicitados e visa desenvolver um entendimento com os fornecedores dos requisitos a respeito de seus significados. À medida que o projeto amadurece e os requisitos vão sendo derivados, todas as atividades ou disciplinas receberão requisitos. Para impedir que os requisitos cresçam indistintamente, critérios são estabelecidos para determinar canais apropriados ou fontes oficiais que devem recebê-los. A admissão de requisitos deve ser analisada com os responsáveis com a finalidade de assegurar um entendimento compatível e compartilhado sobre o significado dos requisitos. Esta análise resulta em um acordo sobre o conjunto de requisitos.

➤ SP1.2 – Obter comprometimento dos requisitos

A segunda SP trata dos acordos e comprometimentos entre os responsáveis por realizar as atividades necessárias para implementar os requisitos, assegurando que os participantes do projeto estejam comprometidos com os atuais requisitos acordados e com as alterações necessárias nos planos de projeto, atividades e produtos de trabalho.

➤ SP1.3 – Gerenciar Mudanças de Requisitos

A terceira SP trata de gerenciar as mudanças nos requisitos à medida que evoluem no projeto, já que em seu andamento, os requisitos mudam, de modo que podem ser incluídos e as mudanças podem ocorrer nos que já existem. Faz-se necessário gerenciar inclusões e

mudanças de forma eficiente. Logicamente, a fonte de cada requisito deve ser conhecida e o fundamento lógico de qualquer mudança deve ser documentado para que se possa analisar efetivamente o impacto das alterações.

➤ SP1.4 – Manter a Rastreabilidade Bidirecional dos Requisitos

Quando se tem uma boa gerência de requisitos, a rastreabilidade pode ser estabelecida desde a fonte até o menor nível do requisito e vice-versa. Esta rastreabilidade bidirecional permite determinar se todos os requisitos de origem foram tratados e se todos os níveis de requisitos podem ser rastreados até um requisito de origem.

➤ SP1.5 – Identificar Inconsistências Entre Trabalho de Projeto e Requisitos

Com o objetivo de não ter inconsistências entre trabalho de projeto e os requisitos, tem-se necessidade de que tais inconsistências sejam identificadas, permitindo promover ações corretivas ao sistema, bem como manter a documentação, incluindo o fundamento lógico, origens e condições.

3.2 MPS-BR

O MPS-BR é um modelo que busca adequar-se ao perfil de organizações de diferentes tamanhos e tipos, mas dá especial atenção às micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software.

Este modelo define sete níveis de maturidade. Este trabalho abordará especificamente o nível G – Parcialmente Gerenciado, que possui como um de seus processos a Gerência de Requisitos. A Figura 6 mostra os níveis do modelo:

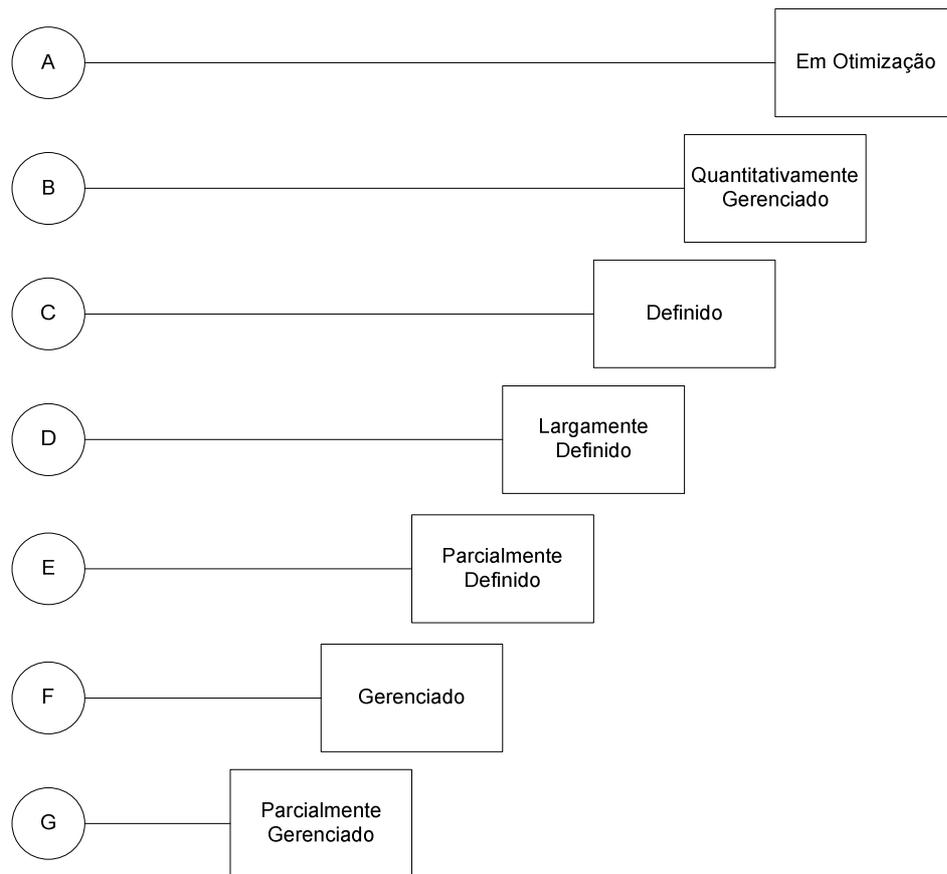


Figura 6. Níveis do MPS-BR

A seguir são apresentadas as definições dos níveis segundo o Guia Geral do MPS-BR (ROCHA; MAGALHÃES, 2007, p. 23):

➤ **Nível G – Parcialmente Gerenciado:**

- **Gerência de Projetos:** visa estabelecer e manter planos que definem atividades recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto;
- **Gerência de Requisitos:** gerenciar os requisitos dos produtos e componentes do produto do projeto e identificar inconsistências entre requisitos, os planos do projeto e os produtos de trabalho do projeto.

➤ Nível F – Gerenciado:

- Aquisição: gerenciar a aquisição de produtos e/ou serviços que satisfaçam a necessidade expressa pelo adquirente.
- Gerência de configuração: estabelecer e manter a integridade de todos os produtos de trabalho de um processo ou projeto e disponibilizá-los a todos os envolvidos.
- Garantia da Qualidade: assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos e recursos predefinidos.
- Medição: coletar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais.

➤ Nível E – Parcialmente Definido:

- Avaliação e Melhoria do Processo Organizacional: determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos.
- Definição do Processo Organizacional: estabelecer e manter um conjunto de ativos de processo organizacional e padrões do ambiente de trabalho usáveis e aplicáveis às necessidades de negócio da organização.
- Gerência de Recursos Humanos: prover a organização e os projetos com os recursos humanos necessários e manter suas competências consistentes com as necessidades do negócio.
- Gerência de Reutilização: gerenciar o ciclo de vida dos ativos reutilizáveis.

➤ Nível D – Largamente Definido:

- Desenvolvimento de Requisitos: estabelecer os requisitos dos componentes do produto, do produto e do cliente.
- Integração do Produto: compor os componentes do produto, produzindo um produto integrado consistente com o projeto, e demonstrar que os requisitos

funcionais e não-funcionais são satisfeitos para o ambiente alvo ou equivalente.

- Projeto e Construção do Produto: projetar, desenvolver e implementar soluções para atender aos requisitos.
- Validação: confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido.
- Verificação: confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados.

➤ Nível C – Definido:

- Análise de Decisão e Resolução: analisar possíveis decisões usando um processo formal, com critérios estabelecidos, para avaliação das alternativas identificadas.
- Desenvolvimento para Reutilização: identificar oportunidades de reutilização sistemática na organização e, se possível, estabelece um programa de reutilização para desenvolver ativos a partir de engenharia de domínios de aplicação.
- Gerência de Riscos: identificar, analisar, tratar, monitorar e reduzir continuamente os riscos em nível organizacional e de projeto.

➤ Nível B – Quantitativamente Gerenciado: é composto pelos processos dos níveis de maturidade anteriores (G ao C), sendo que ao processo Gerência de Projetos são acrescentados novos resultados.

➤ Nível A – Em Otimização:

- Análise de Causas de Problemas e Resolução: identificar causas de defeitos e de outros problemas e tomar ações para prevenir suas ocorrências no futuro.

O processo de Gerência de Requisitos do nível G do MPS-BR tem como objetivo gerenciar os requisitos e os componentes do produto do projeto e identificar inconsistências entre os requisitos, planos do projeto e os produtos de trabalho do projeto. O processo atende a dois atributos do processo: AP1.1 – O processo é executado, que mede quanto o processo

atinge o seu propósito e AP2.1 – O processo é gerenciado, que mede quanto à execução do processo é gerenciada.

Como resultados esperados do processo de Gerência de Requisitos têm-se:

- O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- Os requisitos de software são aprovados utilizando critérios objetivos;
- A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- Mudanças nos requisitos são gerenciadas ao longo do projeto.

3.3 Considerações finais

Os dois modelos de qualidade vistos são de grande importância para atestar a maturidade dos processos das organizações. Nota-se a relevância da gerência de requisitos, onde cada modelo possui uma parte que trata especificamente desta área.

O próximo capítulo irá expor os modelos de qualidade de produto de software, onde serão vistas as normas relacionadas com a avaliação de produtos de software.

4 Modelos de qualidade de produto de software

Atualmente, diversos softwares têm sido desenvolvidos a fim de atender a grande demanda das diversas áreas da economia. Mas, muitos deles são produzidos sem seguir um padrão ou norma, já que os fabricantes estão focados apenas em satisfazer as necessidades iniciais dos clientes, deixando em segundo plano fatores relacionados à manutenção e durabilidade. Com isso, a qualidade desses produtos gera consideráveis preocupações.

Visando a avaliação da qualidade do produto de software, a Organização Internacional para Padronização (ISO), juntamente com a Comissão Eletro-Técnica Internacional (IEC), elaboraram um conjunto de normas a respeito da padronização mundial.

Este capítulo abordará três normas que abordam a avaliação da qualidade de produto de software: a ISO/IEC 9126, relacionada às características de qualidade de produto de software, a ISO/IEC 14598, que serve como guia para avaliação de produto de software e a ISO/IEC 12119, que trata da avaliação de pacotes de software.

4.1 ISO/IEC 9126

A norma ISO/IEC 9126 é formada por um conjunto de aspectos que devem ser observados em um software para que seja verificada sua qualidade. Permite a avaliação do software antes de sua divulgação, observando se atende os requisitos durante seu ciclo de vida de desenvolvimento. Com isso, é possível verificar se o produto é considerado um software de qualidade. Está dividida em quatro partes:

- ISO/IEC 9126-1 – Modelo de Qualidade
- ISO/IEC 9126-2 – Métricas Externas;
- ISO/IEC 9126-3 – Métricas Internas;
- ISO/IEC 9126-4 – Métricas de Qualidade em Uso.

4.1.1 ISO/IEC 9126-1

O primeiro documento da ISO/IEC 9126 apresenta seis características com suas sub-características para o modelo de qualidade de produto de software. São elas:

- Funcionalidade: capacidade de prover funções que, quando o software estiver sendo usado em determinadas condições, atendam as necessidades.
 - Sub-características:
 - Adequação: capacidade de prover funções apropriadas aos objetivos do cliente;
 - Acurácia: capacidade de fornecer resultados com precisão adequada;
 - Interoperabilidade: capacidade de interação com outros sistemas;
 - Segurança de acesso: capacidade de proteger os dados contra acessos não permitidos;
 - Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à funcionalidade.
- Confiabilidade: capacidade de manter o nível de desempenho sob condições estabelecidas durante um período de tempo.
 - Sub-características:
 - Maturidade: capacidade do software de evitar falhas decorrentes de defeitos;
 - Tolerância a falhas: capacidade de manter o funcionamento mesmo na ocorrência de defeitos;
 - Recuperabilidade: capacidade de recuperação após falhas;
 - Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à confiabilidade.

- Usabilidade: evidencia o esforço necessário para utilizar o software por um conjunto de usuários.
 - Sub-características:
 - Inteligibilidade: capacidade de prover facilidade de compreensão das funcionalidades por parte dos usuários;
 - Apreensibilidade: capacidade de fazer o usuário entender o produto;
 - Operacionalidade: capacidade do software de fazer com que o usuário o opere;
 - Atratividade: capacidade de ser atraente para o cliente;
 - Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à usabilidade.
- Eficiência: relaciona o nível de desempenho do software e a quantidade de recursos utilizados sob condições pré-estabelecidas.
 - Sub-características:
 - Comportamento em relação ao tempo: capacidade de prover informações em período de tempo adequado;
 - Comportamento em relação aos recursos: capacidade de usar os recursos adequadamente;
 - Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à eficiência;
- Manutenibilidade: evidencia o esforço necessário para fazer modificações necessárias no software.
 - Sub-características:
 - Analisabilidade: capacidade de reconhecer problemas causados por defeitos;
 - Modificabilidade: capacidade de receber modificações;
 - Estabilidade: capacidade de manter-se estável com as modificações;

- Testabilidade: capacidade de validar as modificações do produto;
- Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à manutenibilidade;
- Portabilidade: capacidade do software de ser transferido de um certo ambiente para outro.
 - Sub-características:
 - Adaptabilidade: capacidade de se adaptar a diferentes ambientes sem auxílio externo;
 - Capacidade para ser instalado: capacidade de ser instalado em um certo ambiente;
 - Co-existência: capacidade de operar com outro sistema no mesmo ambiente;
 - Capacidade para substituir: capacidade de substituir outro sistema com a mesma finalidade;
 - Conformidade: capacidade do produto de software de estar adequado a padrões, normas ou convenções relativos à portabilidade.

A Figura 7 dá uma visão geral da primeira parte da ISO 9126 (KOSCIANSKI, et al, 1999, p.24):

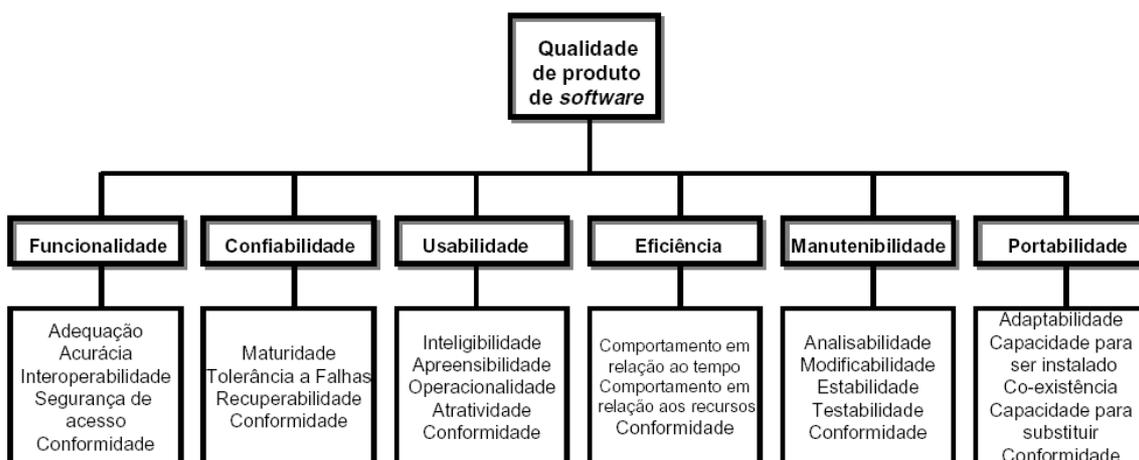


Figura 7. ISO/IEC 9126-1

4.1.2 ISO/IEC 9126-2

A segunda parte da ISO define métricas externas para fazer a medição de qualidade das características e sub-características da primeira parte da norma. Essas métricas são aplicáveis a um software executável tanto na fase de teste quanto no início da operação. A medição pode ser realizada em conformidade com os objetivos esperados do sistema a ser adquirido. A separação das características possibilita a atribuição de pesos, onde se pode definir o grau de importância de certas características.

As métricas externas devem ser utilizadas tanto na avaliação do comportamento do sistema em situações específicas quanto na avaliação e indicação de que o produto é eficaz, levando em consideração as reais necessidades dos clientes em termos de operação.

Alguns exemplos de métricas externas:

- Em relação à adequação: cobertura das funções implantadas;
- À maturidade: resolução de falhas;
- Inteligibilidade: entendimento das entradas e saídas.

4.1.3 ISO/IEC 9126-3

As métricas internas dizem respeito às medições diretas ou indiretas de um produto de software levando em consideração suas próprias características internas, sem que haja necessidade da execução do sistema. Por exemplo: linhas de código, número de erros achados em revisões dentre outros.

Essas métricas permitem ao usuário medir a qualidade dos executáveis durante o processo de desenvolvimento, prevendo assim, a qualidade do produto final. Com isso, o usuário pode identificar problemas de qualidade, corrigindo-os durante o ciclo de desenvolvimento.

Como exemplo, pode-se citar:

- Em relação às métricas de segurança: prevenção da corrupção dos dados;
- Tolerância a falhas: não permissão de operações incorretas;
- Adaptabilidade: adaptabilidade ao ambiente de hardware.

4.1.4 ISO/IEC 9126-4

A quarta e última parte da norma diz respeito à definição de métricas de qualidade em uso, ou seja, valida a qualidade do produto em ambientes e tarefas comuns ao usuário. Os atributos desta parte são organizados pelas suas características. São eles: efetividade, produtividade, segurança e satisfação.

Os usuários podem definir e aplicar métricas para seus próprios ambientes de aplicação.

Pro exemplo, em relação à efetividade, pode-se definir a proporção de tarefas completadas.

Neste trabalho, serão utilizadas a primeira e a segunda parte da norma, ou seja, o modelo de qualidade e as métricas externas.

4.2 ISO/IEC 14598

A série de normas ISO/IEC 14598, que deve ser utilizada juntamente com a ISO/IEC 9126, serve para definir o processo de avaliação de softwares e para fornecer orientações para avaliação prática. Esta avaliação é direcionada tanto para softwares existentes como para os que ainda estão em fase de desenvolvimento. Em termos gerais, mantém o foco nas métricas e no processo de aplicação das mesmas sobre as características e sub-características descritas na ISO/IEC 9126.

A norma é composta por seis documentos, descritos a seguir:

- ISO/IEC 14598-1 Visão Geral: fornece uma visão geral dessa série e dos processos de avaliação, ou seja, define todo o funcionamento da norma. Nela também são definidos os termos utilizados no modelo, bem como conceitos e a execução do processo de avaliação, para ser utilizado por clientes, avaliadores e desenvolvedores. A Figura 8 ilustra esta primeira parte da norma (KOSCIANSKI, et al, 1999, p. 5):



Figura 8. ISO/IEC 14598-1

Como se pode ver, existem várias atividades que por sua vez são divididas em sub-atividades.

Estabelecer o propósito da avaliação visa apoiar diretamente o desenvolvimento e a aquisição de produto de software que atenda as necessidades do cliente. Com isso, assegura-se que o produto possui a qualidade requerida.

Identificar os tipos de produtos a serem avaliados determina o estágio do ciclo de vida que o produto se encontra, o que auxilia na definição de métricas.

Especificar o modelo de qualidade seleciona as características de qualidade relevantes, com isso a qualidade de produto de software é desdobrada em diferentes características.

As três sub-atividades seguintes dizem respeito às métricas, pontuação e critérios. Selecionar métricas faz com que os atributos possam ser definidos com uma métrica. Estabelecer níveis de pontuação para as métricas proporciona um mapeamento em escala dos resultados. Por exemplo, os resultados podem ser satisfatórios ou insatisfatórios. Estabelecer critérios para julgamento define os critérios para as características de qualidade.

Projetar o plano de avaliação descreve os métodos de avaliação e cronograma das ações do avaliador.

Em seguida tem-se a atividade de obter as medidas, onde as métricas selecionadas são aplicadas ao produto de software e obtêm-se os valores nas escalas das métricas. Comparar com critérios compara o valor medido com os critérios pré-determinados. Finalmente a atividade de julgar os resultados gera como resultado a declaração de quanto o software atende os requisitos de qualidade.

- ISO/IEC 14598-2 Planejamento e Gerenciamento: descreve atividades de planejamento e gerenciamento do processo de avaliação apresentando recomendações e orientações para uma função de suporte ao processo. Esse suporte diz respeito ao planejamento e gerenciamento de um processo de avaliação de produto de software bem como a tecnologia necessária neste processo.

- ISO/IEC 14598-3 Processo para a equipe de desenvolvimento: descreve atividades de avaliação durante o processo de desenvolvimento e manutenção de software. Esta parte é utilizada por organizações que querem desenvolver um novo produto ou melhorar um já existente. Foca na seleção e no registro de indicadores que possam ser avaliados, a partir de produtos intermediários, coletados nas fases de desenvolvimento do sistema, visando prever a qualidade do produto final.
- ISO/IEC 14598-4 Processo para o comprador: fornece atividades de avaliação no processo de seleção e avaliação de softwares. Utilizada por clientes que desejam adquirir ou reutilizar produtos de software. Pode ser utilizado para observar a aceitação de um produto ou definir a escolha entre vários produtos existentes.
- ISO/IEC 14598-5 Processo para o avaliador: descreve o ciclo de vida da avaliação, com definição das atividades, incluindo relações entre avaliador e requisitante. Basicamente segue o processo descrito na primeira parte da ISO e detalha as fases deste processo para a função de avaliação.
- ISO/IEC 14598-6 Módulos de avaliação: fornece pacotes estruturados de métodos e ferramentas para apoio das partes relacionadas. Orienta a documentação de módulos de avaliação que devem conter o modelo de qualidade, as informações e dados relativos à aplicação prevista no modelo, bem como informações sobre a real aplicação do modelo.

Resumindo, esta norma permite uma avaliação padronizada das características de qualidade de um software. Diferente da ISO/IEC 9126, esta norma descreve detalhes mínimos, que inclui modelos para relatórios de avaliação, técnicas para medir as características, documentos necessários para avaliação e fases da avaliação.

4.3 ISO/IEC 12119

A ISO/IEC 12119 é uma norma que foi publicada em 1994 e é utilizada na avaliação de pacotes de software do jeito que são liberados para o mercado. Esses pacotes também são conhecidos como “software de prateleira”. Além de estabelecer requisitos para esse tipo de software, a norma provê instruções para o teste dos pacotes.

Pacote de software é a expressão utilizada para referenciar o conjunto completo e documentado de programas fornecidos aos usuários.

A Figura 9 representa a estrutura da norma (NASCIMENTO, et al, s.d., p. 4).

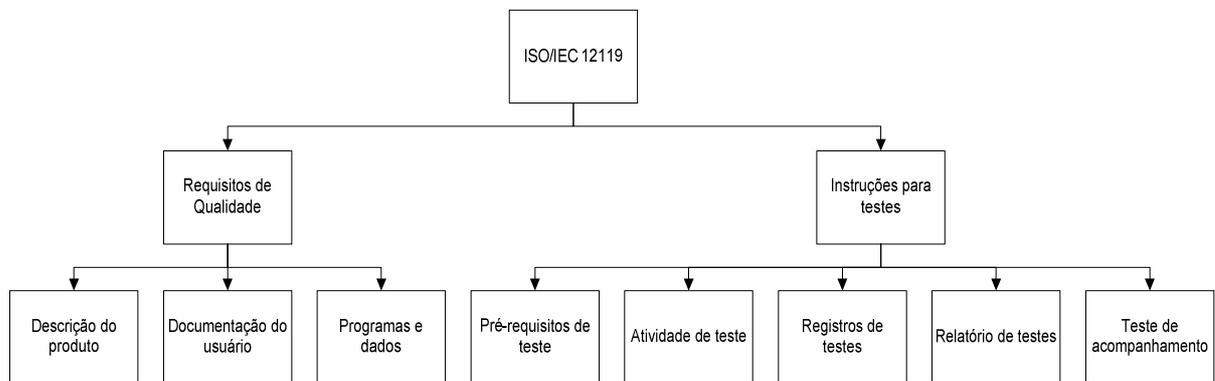


Figura 9. Estrutura da ISO/IEC 12119

4.3.1 Requisitos de Qualidade

Este item se refere à existência da Documentação do Pacote, ou seja, a Descrição do Produto, a Documentação do Usuário e Programas e Dados.

A Descrição do Produto é um documento que mostra as principais características de um pacote de software, visando auxiliar o cliente na adequação do produto aos seus interesses e servir como base de testes. Esse documento deve estar disponível ao usuário através de algum meio (on-line, CD) independente da aquisição ou não do produto. A norma define aspectos que indicam o que deve estar contido na descrição e propõe uma linguagem clara e compreensível ao cliente.

A Documentação do Usuário é um conjunto de documentos também disponíveis por algum meio ao usuário, que se refere à utilização do produto. Deve incluir informações sobre a instalação, o uso em si da aplicação e sobre a manutenção do software.

Programas e Dados descreve em detalhes cada uma das funções do programa. Utiliza as mesmas definições da norma ISO/IEC 9126. Assim, inclui declarações a respeito de Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade, com destaque às três primeiras.

4.3.2 Instruções para Testes

O item Instruções para Testes basicamente recomenda a forma com que o produto de software deve ser testado em relação aos requisitos de qualidade.

Pré-requisitos de Teste possui três componentes: presença de Itens, onde devem estar presentes, para execução do teste, todos os componentes a serem entregues e os documentos de requisitos identificados na Descrição do Produto; Presença de Componentes do Sistema, que deve estar disponível todo ambiente de hardware e software identificados na descrição do produto; Treinamento, onde se for mencionado na Descrição do Produto, o responsável pelo teste deve ter acesso ao material e ao programa de treinamento.

Atividades de Teste também possui três componentes: Descrição do Produto, Documentação do Usuário e Programas e Dados, onde todo o requisito especificado em cada parte deve ser testado.

Registros de Teste propõe que os registros devem conter informações suficientes para que se possa repetir o teste.

No Relatório de Teste deve ter um resumo com os objetivos e resultados dos testes realizados.

O Teste de Acompanhamento afirma que todas as partes modificadas, quando o produto é testado novamente, devem ser testadas como um produto novo.

4.4 Considerações finais

Os modelos de qualidade expostos permitem avaliar com exatidão a qualidade dos produtos de software, fornecendo poderosa ferramenta para esse fim. Primeiramente foi vista a ISO 9126 que fornece um conjunto de aspectos que devem ser observados em um software para que seja verificada sua qualidade. Posteriormente foi apresentada a ISO 14598, que serve para definir o processo de avaliação de softwares e para fornecer orientações para avaliação prática. Finalizando abordou-se a ISO 12119 que é utilizada na avaliação de pacotes de software do jeito que são liberados para o mercado

O capítulo a seguir irá abordar o sistema de avaliação das ferramentas selecionadas.

5 Processo de avaliação

Este capítulo apresenta o sistema de avaliação adotado para as ferramentas de Gerência de Requisitos.

Primeiramente foi feito um estudo na Internet a respeito das principais ferramentas gratuitas utilizadas na gerência de requisitos. Esse estudo foi baseado em sites e fóruns especializados em Engenharia de Software, que têm credibilidade reconhecida na área.

Após o estudo inicial, as ferramentas selecionadas (que serão descritas mais a frente) foram baixadas e instaladas para análise.

No processo de avaliação foram definidas as características, sub-características e atributos que devem ser avaliados. A avaliação foi realizada através da adoção de métricas pré-estabelecidas. Basicamente, cada característica é desmembrada em sub-características, e essas por sua vez são compostas por atributos. Assim, as sub-características terão suas notas quando todos seus atributos forem avaliados. Da mesma forma as características terão suas notas assim que o conjunto de sub-características forem avaliados. Este processo dá idéia de uma árvore invertida, onde as folhas são os atributos, os pais das folhas são as sub-características e os pais das sub-características são as características, como mostra a Figura 10.

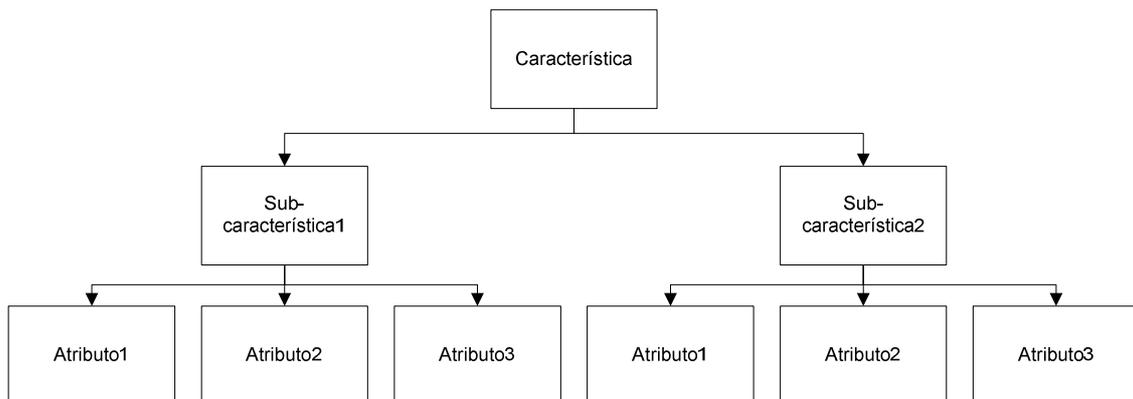


Figura 10. Distribuição dos atributos e sub-características de uma característica

As notas finais da avaliação de cada software serão dadas pela soma das notas das características avaliadas. Quanto maior a nota final, maior será a “qualidade do produto”.

5.1 Usuários

Este item tem como objetivo definir o perfil do usuário do software a ser avaliado.

A ISO/IEC 9126 especifica três tipos de usuários: gerentes de desenvolvimento, desenvolvedores e operadores.

O perfil adotado neste trabalho é do **operador**, ou seja, uma pessoa responsável por operar a ferramenta, realizando atividades como inserção de requisitos, modificação dentre outras.

5.2 Características, sub-características e atributos

Com a finalidade de avaliar as ferramentas, foi definido um modelo misto que une características e sub-características baseados na ISO/IEC 9126 com requisitos da ISO/IEC 12119. Um conjunto de atributos (representado pelas perguntas) foi definido para cada sub-característica.

Da ISO/IEC 9126:

- Funcionalidade:
 - Adequação:
 - Todas as funções necessárias foram implementadas?
 - Neste atributo analisa-se: importação e exportação de documentos, armazenamento, busca de requisitos, gerenciamento de mudanças, rastreamento e geração de relatórios.

- Todas as funções estão corretas?
 - Verifica-se se as funções estão corretamente implementadas e fornecem o resultado esperado.
- Acurácia:
 - Os dados são representados de forma satisfatória?
 - Analisa-se a apresentação dos resultados em termos estéticos e de completude das informações.
- Segurança de acesso:
 - Há acesso com login e senha?
 - Verifica-se a segurança dos dados com autenticação dos usuários.
 - Existe um controle de acesso a funções de acordo com cada usuário?
 - Analisa-se a diferenciação entre os níveis de usuários e as ações permitidas a cada um deles.
- Confiabilidade:
 - Maturidade:
 - Mensagens alertam operações perigosas?
 - Verifica-se a presença de mensagens que mostram que alguma ação perigosa foi acionada. Por exemplo a exclusão de artefatos.
 - Mensagens de erro explicam claramente sua causa?
 - As mensagens de erro explicam de forma suficiente e clara a ocorrência de erros, como criação de requisitos com falta de informações.
 - Recuperabilidade:
 - Existe a função “desfazer”?
 - Importante recurso de desfazer ações realizadas após verificação de equívoco.

➤ Usabilidade:

▪ Intelligibilidade:

- Existem “Assistentes” para as funções?
 - Verifica-se a presença de assistentes que guiam o uso das funções.
- Os menus estão bem organizados?
 - Forma de prover melhor entendimento das funções.

▪ Aprendibilidade:

- Existem menus pop-up com frases explicativas das funções?
 - Presença de pequenos quadros que surgem na tela mostrando a funcionalidade referida.
- Os desenhos das funções estão de acordo com suas finalidades?
 - Estuda-se se as gravuras das funções proporcionam um entendimento claro de sua finalidade.

▪ Atratividade:

- Objetos bem distribuídos de acordo com a função?
 - Este item diz respeito à organização dos recursos na tela.
- É possível customizar?
 - Verifica a possibilidade de o usuário alterar a aparência do sistema, conforme suas necessidades.

➤ Portabilidade:

▪ Adaptabilidade:

- Roda em diferentes Sistemas Operacionais?
 - Estuda-se a possibilidade de instalação em diferentes plataformas.

▪ Capacidade para ser instalado:

- A instalação é automática?
 - Analisa-se se recursos externos foram necessários durante a instalação.

Da ISO/IEC 12119:

- Documentação:
 - Descrição do produto:
 - Possui uma descrição do produto?
 - Analisa-se a existência de textos descrevendo o produto.
 - Documentação do usuário:
 - Possui documentação do usuário (manual de instalação, manual de uso)?
 - Verifica-se a existência de manuais de instalação e de uso.
 - Programas e dados:
 - Possui descrição das funcionalidades?
 - Verifica-se a existência de um texto com a descrição das funcionalidades do sistema.

5.3 Distribuição dos pontos

Com a finalidade de fazer uma avaliação consistente e não-tendenciosa, foi feita a seguinte distribuição de pontos: para cada software foi definido uma pontuação total de 120 pontos. Esses 120 pontos foram distribuídos entre as características que, como são cinco, cada uma ficou com 24 pontos. Da mesma forma, os 24 pontos foram distribuídos igualmente entre as sub-características, que por sua vez foram distribuídos igualmente entre os atributos.

A Tabela 1 apresenta a divisão de notas:

Característica, Sub-característica e Atributo	Nota total por item
➤ Funcionalidade:	24
▪ Adequação:	8
• Todas as funções necessárias foram implementadas?	4
• Todas as funções estão corretas?	4
▪ Acurácia:	8
• Os dados são representados de forma satisfatória?	8
▪ Segurança de acesso:	8
• Há acesso com login e senha?	4
• Existe um controle de acesso a funções de acordo com cada usuário?	4
➤ Confiabilidade:	24
▪ Maturidade:	12
• Mensagens alertam operações perigosas?	6
• Mensagens de erro explicam claramente sua causa?	6
▪ Recuperabilidade:	12
• Existe a função “desfazer”?	12
➤ Usabilidade:	24
▪ Inteligibilidade:	8
• Existem “Assistentes” para as funções?	4
• Os menus estão bem organizados?	4
▪ Aprendibilidade:	8
• Existem menus pop-up com frases explicativas das funções?	4
• Os desenhos das funções estão de acordo com suas finalidades?	4
▪ Atratividade:	8
• Objetos bem distribuídos de acordo com a função?	4
• É possível customizar?	4
➤ Portabilidade:	24
▪ Adaptabilidade:	12
• Roda em diferentes Sistemas Operacionais?	12
▪ Capacidade para ser instalado:	12
• A instalação é automática?	12
➤ Documentação:	24
▪ Descrição do produto:	8
• Possui uma descrição do produto?	8
▪ Documentação do usuário:	8
• Possui documentação do usuário (manual de instalação, manual de uso)?	8
▪ Programas e dados:	8
• Possui descrição das funcionalidades?	8

Tabela 1. Distribuição dos pontos

5.4 Sistema de métricas adotado

A precisão de uma avaliação de qualidade depende em grande parte das métricas escolhidas [6]. Com este pensamento e baseado nos requisitos de avaliação da ISO/IEC 14598-1 e da ISO/IEC 9126-1, o seguinte sistema de métricas foi definido:

- 0 – Não possui: Significa que o atributo avaliado está completamente ausente;
- 0,5 – Possui parcialmente: O atributo avaliado possui está presente parcialmente. Com isso, a Nota total por item será dividida pela metade;
- 1 – Possui totalmente: O atributo avaliado está totalmente presente;

Vale ressaltar que esse sistema de métricas é utilizado na avaliação dos **atributos**, já que a nota das sub-características é dada pela soma das notas dos atributos. Da mesma forma, a nota das características é dada pela soma das notas das sub-características.

5.4.1 Cálculo da nota final

Cada software será primeiramente analisado separadamente dos outros. Assim, será possível observar as notas de cada item isoladamente. A nota final de cada software se dará da seguinte forma:

$$\text{Nota final} = \sum (\text{Nota da característica})$$

Por sua vez, a nota de cada característica será dada por:

$$\text{Nota da característica} = \sum (\text{Nota da sub-característica})$$

Da mesma forma, a nota das sub-características será dada por:

$$\text{Nota da sub-característica} = \sum (\text{Nota do atributo} * \text{Nota total por item})$$

Como foi dito, quanto maior a nota final de cada software, maior será sua qualidade, baseada nos quesitos levados em consideração.

Com a nota final, será possível fazer a comparação entre os produtos analisados.

5.5 Considerações finais

Com este capítulo, pode-se ver como foi feito o sistema de avaliação dos programas selecionados. Após definir o tipo de usuário, foram definidas as características, sub-características e atributos a serem observados, seguidos da distribuição dos pontos. Finalizando foi explicado o sistema de métricas adotado.

O próximo capítulo irá mostrar a avaliação em si das ferramentas.

6 Análise das ferramentas selecionadas

Esta seção irá expor os softwares analisados. Como foi dito, os softwares foram selecionados após pesquisas na Internet, onde foram observadas as ferramentas mais relevantes. Segue a lista de programas:

6.1 Jeremia

Jeremia é uma ferramenta ajuda no desenvolvimento de sistemas com o rastreamento das mudanças de requisitos durante todo o ciclo de vida. Com ele, é possível capturar, analisar e classificar os requisitos. Também é possível fazer link entre os requisitos, o que permite ter um rastreamento eficaz.

O software conta com um conjunto de templates padrões orientados a processos. Os requisitos coletados são inseridos nesses templates automaticamente. É possível exportar documentação em formato XML-DOCBOOK. Utiliza como banco de dados o MYSQL ou ORACLE, onde armazena a documentação e os requisitos.

O programa foi desenvolvido em Java e está sob a licença GPL – General Public License.

A Figura 11 mostra a tela principal do programa.

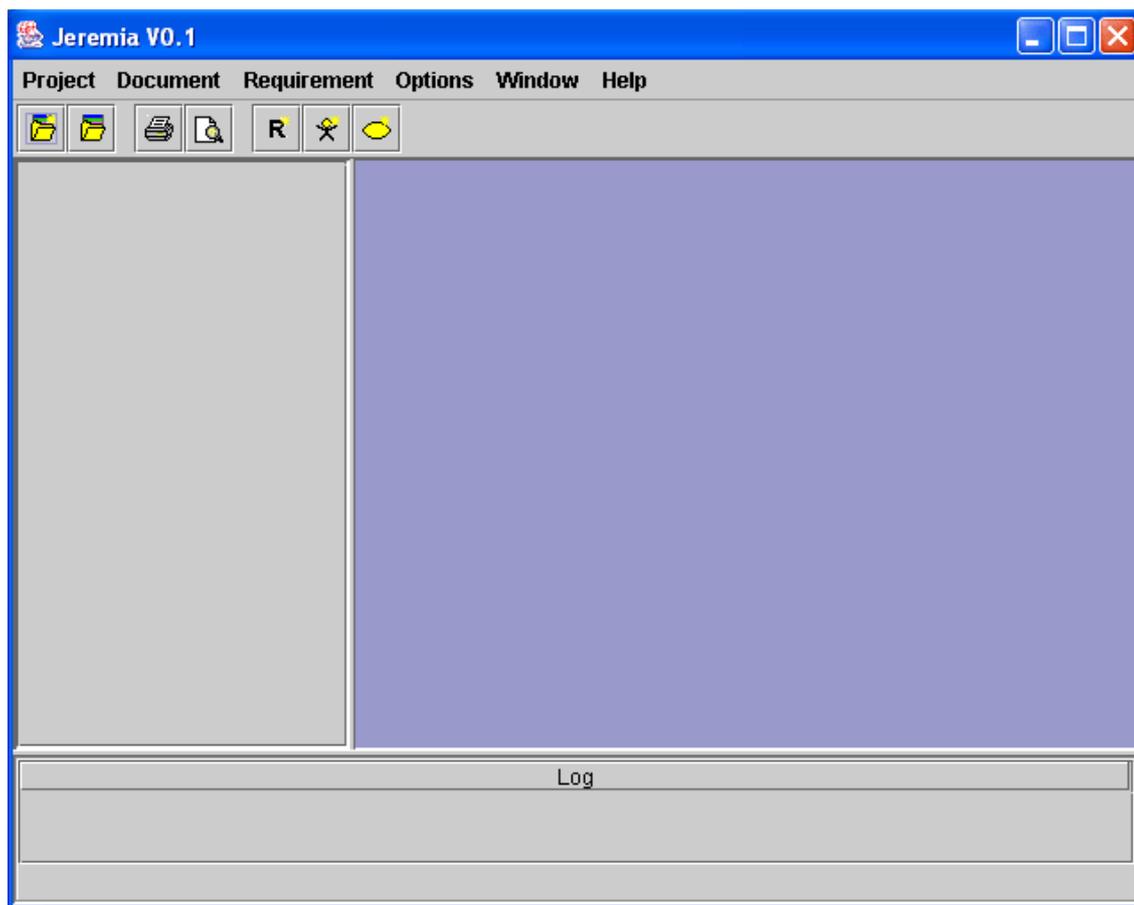


Figura 11. Tela inicial do Jeremia

6.2 OSRMT

O OSRMT (Open Source Requirements Management Tool) é uma ferramenta que foi desenhada para permitir total rastreabilidade dos requisitos. Provê boa gerência dos requisitos bem como dos componentes de produtos de trabalho.

Para iniciá-lo, basta ativar o Servidor e depois o Cliente.

Foi desenvolvido em Java e está sob a licença GPL – General Public License.

A Figura 12 apresenta a tela inicial da ferramenta:

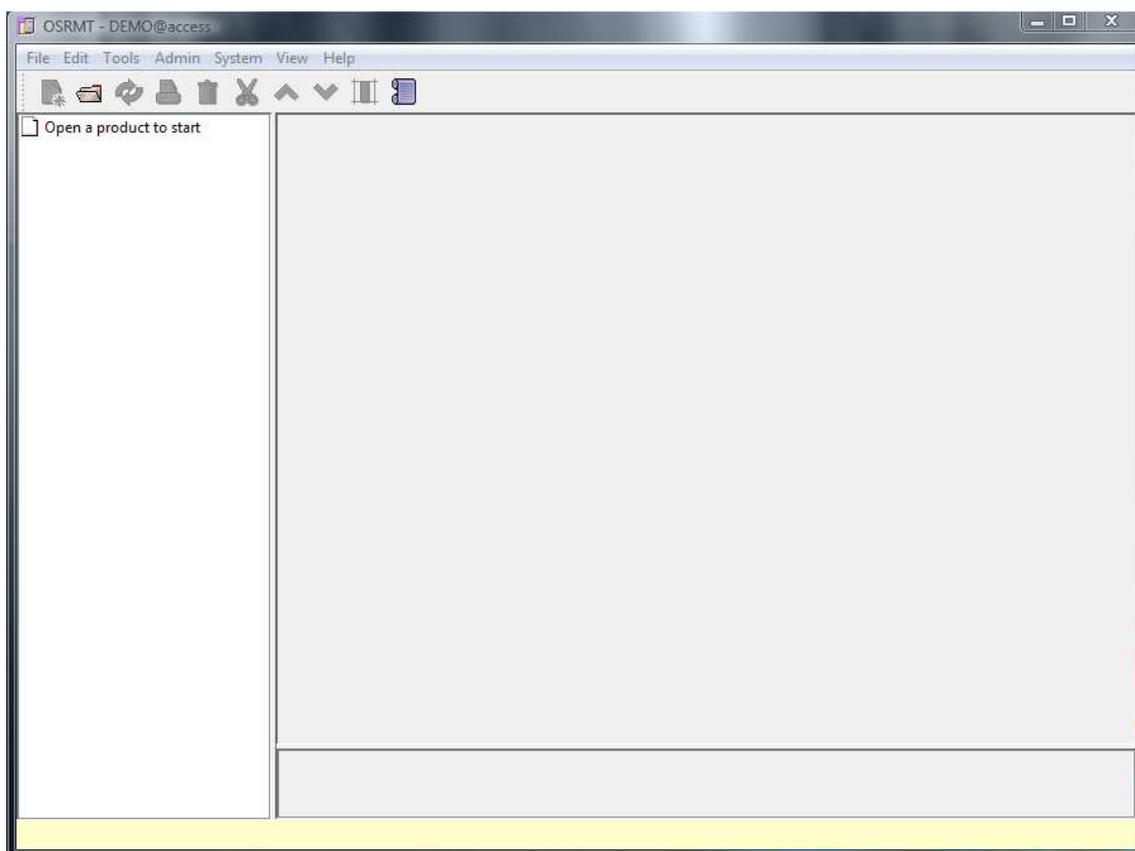


Figura 12. Tela inicial do OSRMT

6.3 Tiger Pro

Desenvolvida pela SEEC (Systems Engineering & Evaluation Centre), TIGER PRO (Tool to InGest and Elucidate Requirements PROfessional) é uma boa ferramenta Freeware que permite adicionar requisitos através de documentos ou diretamente no programa. Produz um sumário de informações na elucidação de requisitos, alocação de prioridades, definição de riscos e estimativa de custo em formato de texto ou gráfico. Licenciado para uso educacional, mas, com autorização do responsável, é possível usá-lo comercialmente.

A Figura 13 mostra a tela inicial do programa:

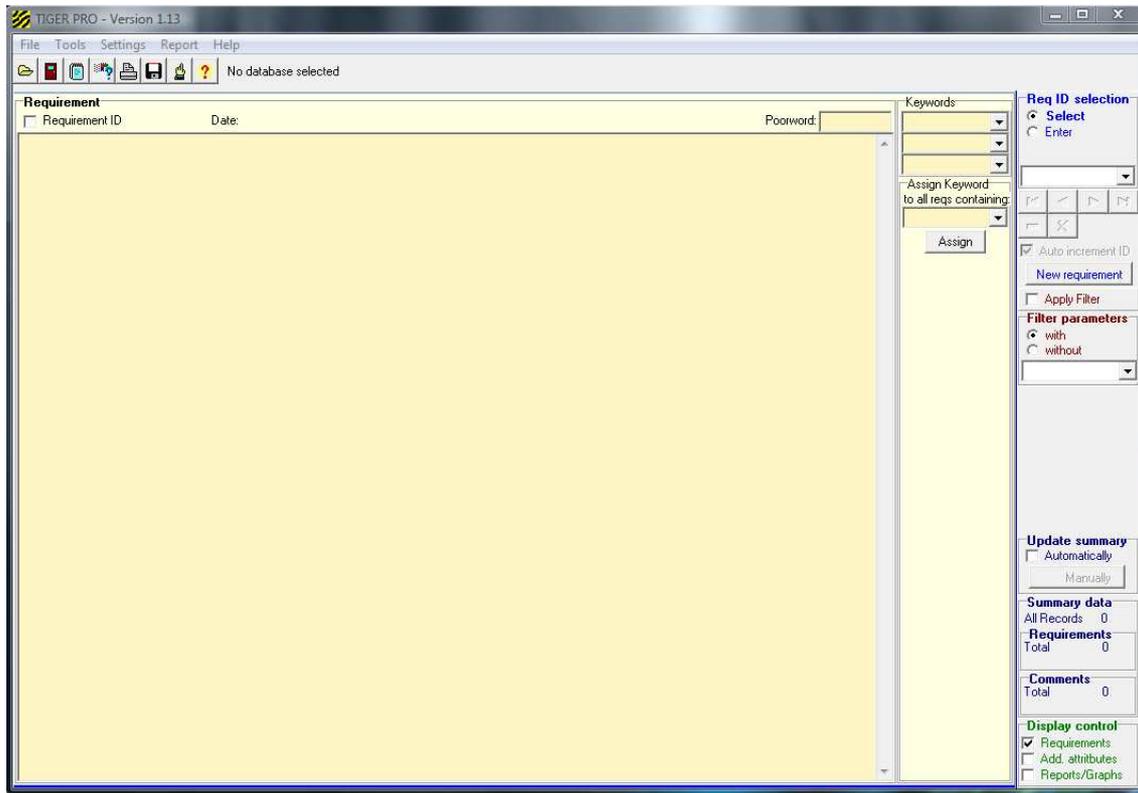


Figura 13. Tela inicial do Tiger Pro

6.4 Xuse

Xuse é uma ferramenta utilizada no browser e visa gerenciar requisitos, casos de uso e outros artefatos no design de softwares. Os comandos são executáveis no em linhas de comando no prompt. Seu foco é uma documentação e comunicação claras.

Está sob a Licença Artística.

A Figura 14 mostra a tela inicial da aplicação:

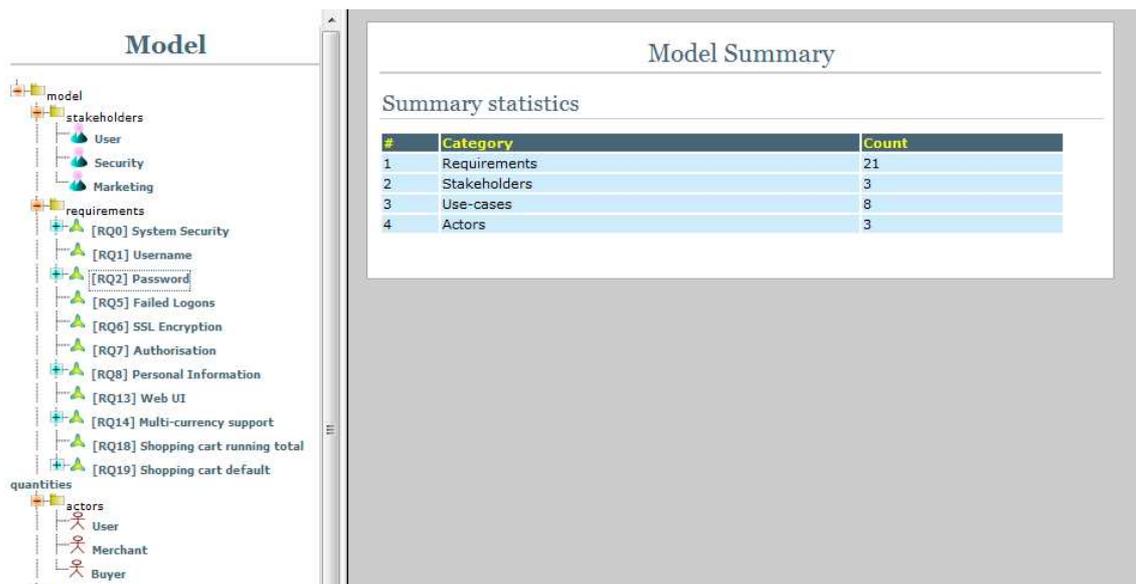


Figura 14. Tela inicial do Xuse

6.5 Análise das ferramentas

Esta seção tem como objetivo a análise em si das ferramentas. Ressalta-se que primeiramente será feito um estudo individual das ferramentas e posteriormente um estudo comparativo entre as ferramentas. O item 6.3 traz uma explicação detalhada do que foi observado em cada programa

Primeiramente será feito o estudo da ferramenta Jeremia, seguido pelos programas OSRMT, Tiger Pro e Xuse.

6.5.1 Jeremia

A Tabela 2 apresenta o estudo realizado sobre a ferramenta Jeremia.

Característica, Sub-característica e Atributo	Nota total por item	Avaliação
➤ Funcionalidade:	24	24
▪ Adequação:	8	8
• Todas as funções necessárias foram implementadas?	4	1
• Todas as funções estão corretas?	4	1
▪ Acurácia:	8	8
• Os dados são representados de forma satisfatória?	8	1
▪ Segurança de acesso:	8	8
• Há acesso com login e senha?	4	1
• Existe um controle de acesso a funções de acordo com cada usuário?	4	1
➤ Confiabilidade:	24	9
▪ Maturidade:	12	9
• Mensagens alertam operações perigosas?	6	1
• Mensagens de erro explicam claramente sua causa?	6	0,5
▪ Recuperabilidade:	12	0
• Existe a função “desfazer”?	12	0
➤ Usabilidade:	24	16
▪ Inteligibilidade:	8	4
• Existem “Assistentes” para as funções?	4	0
• Os menus estão bem organizados?	4	1
▪ Aprendibilidade:	8	8
• Existem menus pop-up com frases explicativas das funções?	4	1
• Os desenhos das funções estão de acordo com suas finalidades?	4	1
▪ Atratividade:	8	4
• Objetos bem distribuídos de acordo com a função?	4	1
• É possível customizar?	4	0
➤ Portabilidade:	24	18
▪ Adaptabilidade:	12	12
• Roda em diferentes Sistemas Operacionais?	12	1
▪ Capacidade para ser instalado:	12	6
• A instalação é automática?	12	0,5
➤ Documentação:	24	16
▪ Descrição do produto:	8	8
• Possui uma descrição do produto?	8	1
▪ Documentação do usuário:	8	8
• Possui documentação do usuário (manual de instalação, manual de uso)?	8	1
▪ Programas e dados:	8	0
• Possui descrição das funcionalidades?	8	0
TOTAL	120	83

Tabela 2. Tabela de análise do Jeremia

6.1.2 OSRMT

A Tabela 3 mostra o resultado do estudo realizado sobre a ferramenta OSRMT.

Característica, Sub-característica e Atributo	Nota total por item	Avaliação
➤ Funcionalidade:	24	24
▪ Adequação:	8	8
• Todas as funções necessárias foram implementadas?	4	1
• Todas as funções estão corretas?	4	1
▪ Acurácia:	8	8
• Os dados são representados de forma satisfatória?	8	1
▪ Segurança de acesso:	8	8
• Há acesso com login e senha?	4	1
• Existe um controle de acesso a funções de acordo com cada usuário?	4	1
➤ Confiabilidade:	24	12
▪ Maturidade:	12	12
• Mensagens alertam operações perigosas?	6	1
• Mensagens de erro explicam claramente sua causa?	6	1
▪ Recuperabilidade:	12	0
• Existe a função “desfazer”?	12	0
➤ Usabilidade:	24	16
▪ Inteligibilidade:	8	4
• Existem “Assistentes” para as funções?	4	0
• Os menus estão bem organizados?	4	1
▪ Aprendibilidade:	8	8
• Existem menus pop-up com frases explicativas das funções?	4	1
• Os desenhos das funções estão de acordo com suas finalidades?	4	1
▪ Atratividade:	8	4
• Objetos bem distribuídos de acordo com a função?	4	1
• É possível customizar?	4	0
➤ Portabilidade:	24	24
▪ Adaptabilidade:	12	12
• Roda em diferentes Sistemas Operacionais?	12	1
▪ Capacidade para ser instalado:	12	12
• A instalação é automática?	12	1
➤ Documentação:	24	24
▪ Descrição do produto:	8	8
• Possui uma descrição do produto?	8	1
▪ Documentação do usuário:	8	8
• Possui documentação do usuário (manual de instalação, manual de uso)?	8	1
▪ Programas e dados:	8	8
• Possui descrição das funcionalidades?	8	1
TOTAL	120	100

Tabela 3. Tabela de análise do OSRMT

6.1.3 Tiger Pro

A Tabela 4 apresenta a avaliação realizada sobre o programa Tiger Pro.

Característica, Sub-característica e Atributo	Nota total por item	Avaliação
➤ Funcionalidade:	24	10
▪ Adequação:	8	6
• Todas as funções necessárias foram implementadas?	4	0,5
• Todas as funções estão corretas?	4	1
▪ Acurácia:	8	4
• Os dados são representados de forma satisfatória?	8	0,5
▪ Segurança de acesso:	8	0
• Há acesso com login e senha?	4	0
• Existe um controle de acesso a funções de acordo com cada usuário?	4	0
➤ Confiabilidade:	24	9
▪ Maturidade:	12	9
• Mensagens alertam operações perigosas?	6	1
• Mensagens de erro explicam claramente sua causa?	6	0,5
▪ Recuperabilidade:	12	0
• Existe a função “desfazer”?	12	0
➤ Usabilidade:	24	10
▪ Inteligibilidade:	8	2
• Existem “Assistentes” para as funções?	4	0
• Os menus estão bem organizados?	4	0,5
▪ Aprendibilidade:	8	6
• Existem menus pop-up com frases explicativas das funções?	4	1
• Os desenhos das funções estão de acordo com suas finalidades?	4	0,5
▪ Atratividade:	8	2
• Objetos bem distribuídos de acordo com a função?	4	0,5
• É possível customizar?	4	0
➤ Portabilidade:	24	12
▪ Adaptabilidade:	12	0
• Roda em diferentes Sistemas Operacionais?	12	0
▪ Capacidade para ser instalado:	12	12
• A instalação é automática?	12	1
➤ Documentação:	24	12
▪ Descrição do produto:	8	8
• Possui uma descrição do produto?	8	1
▪ Documentação do usuário:	8	0
• Possui documentação do usuário (manual de instalação, manual de uso)?	8	0
▪ Programas e dados:	8	4
• Possui descrição das funcionalidades?	8	0,5
TOTAL	120	53

Tabela 4. Tabela de análise do Tiger Pro

6.1.4 Xuse

A Tabela 5 mostra o estudo realizado sobre a ferramenta Xuse.

Característica, Sub-característica e Atributo	Nota total por item	Avaliação
➤ Funcionalidade:	24	14
▪ Adequação:	8	6
• Todas as funções necessárias foram implementadas?	4	0,5
• Todas as funções estão corretas?	4	1
▪ Acurácia:	8	8
• Os dados são representados de forma satisfatória?	8	1
▪ Segurança de acesso:	8	0
• Há acesso com login e senha?	4	0
• Existe um controle de acesso a funções de acordo com cada usuário?	4	0
➤ Confiabilidade:	24	0
▪ Maturidade:	12	0
• Mensagens alertam operações perigosas?	6	0
• Mensagens de erro explicam claramente sua causa?	6	0
▪ Recuperabilidade:	12	0
• Existe a função “desfazer”?	12	0
➤ Usabilidade:	24	4
▪ Inteligibilidade:	8	0
• Existem “Assistentes” para as funções?	4	0
• Os menus estão bem organizados?	4	0
▪ Aprendibilidade:	8	0
• Existem menus pop-up com frases explicativas das funções?	4	0
• Os desenhos das funções estão de acordo com suas finalidades?	4	0
▪ Atratividade:	8	4
• Objetos bem distribuídos de acordo com a função?	4	0
• É possível customizar?	4	1
➤ Portabilidade:	24	24
▪ Adaptabilidade:	12	12
• Roda em diferentes Sistemas Operacionais?	12	1
▪ Capacidade para ser instalado:	12	12
• A instalação é automática?	12	1
➤ Documentação:	24	24
▪ Descrição do produto:	8	8
• Possui uma descrição do produto?	8	1
▪ Documentação do usuário:	8	8
• Possui documentação do usuário (manual de instalação, manual de uso)?	8	1
▪ Programas e dados:	8	8
• Possui descrição das funcionalidades?	8	1
TOTAL	120	66

Tabela 5. Tabela de análise do Xuse

Finalizado o estudo individual das ferramentas, é mostrada uma tabela comparativa onde é possível verificar as diferenças entre as características, podendo servir de apoio na escolha da ferramenta mais adequada.

6.2 Comparação entre as ferramentas

A Tabela 6 faz uma comparação dos resultados obtidos considerando as características estudadas:

	Jeremia	OSRMT	Tiger Pro	Xuse
Funcionalidade	24	24	10	14
Confiabilidade	9	12	9	0
Usabilidade	16	16	10	4
Portabilidade	18	24	12	18
Documentação	16	24	12	24
TOTAL	83	100	53	66

Tabela 6. Tabela de comparação das ferramentas

6.3 Análise dos resultados

Nesta seção são analisados os resultados obtidos na avaliação dos programas.

6.3.1 Jeremia

A ferramenta alcançou resultados satisfatórios na avaliação realizada. Na característica funcionalidade pode-se notar que na ferramenta todas as funções consideradas neste trabalho e descritas no item 5.2 foram implementadas, apresentando resultados corretos e de forma satisfatória.

O programa possui sistema de login e senha, onde cada usuário, de acordo com suas atribuições, tem acesso a funções pertinentes.

Foram observados três tipos de usuários: administrador, que tem acesso a todas as funções do sistema; usuário geral, que tem acesso a várias funções, menos gerencia usuários e alterar o setup; convidado, que apenas pode visualizar requisitos e documentos.

Na característica confiabilidade, foram observadas mensagens alertando operações perigosas, como exclusão de requisitos. Não há o recurso de desfazer ações. Sobre a usabilidade, pode-se notar um bom foco no usuário, ou seja, menus bem organizados, com desenhos intuitivos e com frases com o nome dos recursos.

A respeito da portabilidade, há necessidade de ter instalado o Java e o MySQL, bem como ter conhecimentos de alteração de variáveis de ambiente do Windows. Vale destacar que é uma ferramenta independente de plataforma, podendo ser instalada em diferentes sistemas operacionais. No site da ferramenta encontra-se a descrição do produto, manual de instalação e uso, mas não a descrição das funcionalidades.

6.3.2 OSRMT

Esta ferramenta alcançou ótimos resultados. Todas as funções descritas no item 5.2 foram implementadas, apresentando resultados corretos e bem apresentáveis. Possui login e senha, onde cada usuário tem acesso a funções de acordo com suas atribuições. São eles: administrador, analista de negócio, quadro de controle de mudanças, desenvolvedores, documentadores, gerente de produto, regulador de qualidade e testadores de qualidade.

Mensagens alertam para operações perigosas, como exclusão de artefatos e fornecem informações suficientes ao entendimento. O recurso desfazer não foi observado. Da mesma forma que a ferramenta anterior, o produto foi desenvolvido com foco no usuário, onde os menus são bem organizados, com desenhos fáceis de entender com frases explicando a função.

A instalação foi automática, necessitando de Java e do banco de dados. Da mesma forma que a ferramenta anterior, é uma ferramenta independente de plataforma, podendo ser instalada em diferentes sistemas operacionais. Juntamente com a ferramenta encontra-se o manual de uso e instalação bem como a descrição das funções e do produto. Vale ressaltar que o projeto OSRMT está ativo e atualizações são constantemente elaboradas, o que proporciona um software cada vez melhor.

6.3.3 TigerPro

A análise do software TigerPro não obteve resultados tão relevantes quanto das duas ferramentas anteriores. Como foi dito, é um software com fins educativos, mas com a devida autorização de seus idealizadores, pode ser utilizado para fins comerciais. Sobre a característica funcionalidade, não foram observadas algumas funções definidas no item 5.2, como o gerenciamento de mudanças, recurso de busca de artefatos nem a rastreabilidade dos requisitos. Sendo assim, obteve nota 0,5 na avaliação da adequação. Porém, todas as funções presentes estão corretas.

Devido à grande quantidade de informações na tela, a representação dos dados fica confusa. Não possui login e senha, e não tem diferenciação de usuários, causando uma falha de segurança do programa. Apresenta mensagens indicando atividades perigosas, porém não informa muitos detalhes, recebendo assim a nota 0,5 neste atributo. O recurso desfazer não foi observado. A organização dos menus deixou a desejar, pois há muita informação na tela. Da mesma forma, os desenhos das funções não são muito intuitivos, mas a presença de menus pop-up permitem seu entendimento.

O excesso de informação na tela também prejudica a distribuição dos recursos de acordo com as finalidades, recebendo assim, nota 0,5. A Figura 16 dá um exemplo do excesso. A instalação do programa foi automática, não necessitando de recursos extras, porém, só é aplicável ao ambiente Windows.

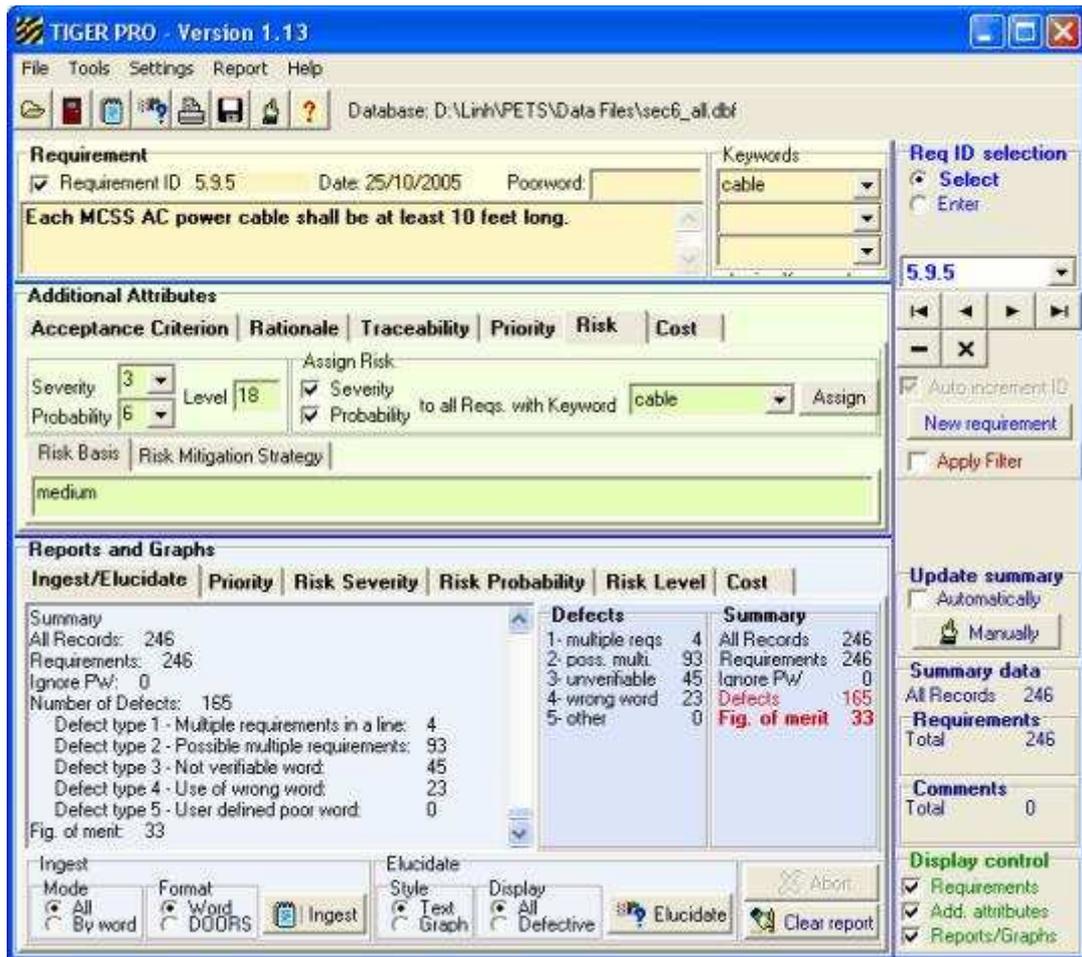


Figura 15. Funcionamento do TigerPro

Em termos de documentação, foi observada a descrição do produto e uma breve descrição de algumas funcionalidades.

6.3.4 Xuse

O projeto da ferramenta Xuse está ativo e constantemente atualizações são lançadas, adicionando funções ao sistema. Porém, atualmente não se apresenta como uma ferramenta dinâmica para o mercado considerando os concorrentes. Em termos de funcionalidade, não foram observados os recursos de busca por requisitos provê gerenciamento de mudanças,

recebendo assim nota 0,5 no atributo adequação. Porém, todas as funções estão corretas e os dados têm representação clara e satisfatória.

Não foram observados os recursos de login e senha, nem a diferenciação de usuários, o que provoca uma falha de segurança do sistema. Não foram observadas mensagens de alerta de operações perigosas e não possui o recurso “desfazer”. Não existem assistentes para as funções e como não há menus, recebeu nota 0 nesses atributos. Da mesma forma os atributos seguintes a respeito de atratividade. Também é possível utilizar em diferentes plataformas. A instalação foi automática, não necessitando de recursos extras. Possui toda a documentação estudada.

6.4 Considerações finais

Com a análise dos programas selecionados pode-se notar a superioridade da ferramenta OSRMT. Como foi dito, o sistema está em constante melhoramento e possui uma comunidade grande de usuários e apoiadores. Nesta ferramenta se destacou a qualidade da documentação e a completude do sistema, proporcionando bom aproveitamento dos recursos.

Porém, convém ressaltar que de acordo com as necessidades do usuário, os outros programas podem atender, levando em conta as diferenças entre as características analisadas, ou seja, pode-se escolher a ferramenta que mais provê portabilidade, ou então que tenha uma documentação mais completa.

7 Conclusão

As ferramentas de Gerência de Requisitos dão grande apoio ao processo de Engenharia de Software. Os sistemas atuais têm necessitado cada vez mais de novos recursos e com isso, um número maior de requisitos. Então, um software para apoiar esse processo se torna um grande aliado aos responsáveis pelos projetos.

Este trabalho pretende servir como apoio à escolha da ferramenta mais apropriada as necessidades dos interessados. Para isso, no capítulo 2 foram introduzidos conceitos de Engenharia de Requisitos, onde seus processos caminham em paralelo com a atividade de Gerência de Requisitos.

No capítulo 3, foi vista a Gerência de Requisitos em Modelos de Qualidade, onde foi possível ter uma visão mais detalhada do CMMI e do MPS-BR.

Posteriormente, no capítulo 4 foram expostas as normas ISO 9126, formada por um conjunto de aspectos que devem ser observados em um software para que seja verificada sua qualidade; ISO 14598, que define o processo de avaliação de softwares e fornece orientações para avaliação prática; e a ISO 12119, utilizada na avaliação de pacotes de software do jeito que são liberados para o mercado.

O capítulo 5 apresentou o processo de avaliação das ferramentas, onde foram definidas as características, sub-características e atributos a serem observados, bem como o sistema de métricas utilizado.

O capítulo 6 mostrou o estudo em si das ferramentas, onde os programas selecionados foram apresentados e submetidos ao processo de avaliação.

7.1 Dificuldades encontradas

As principais dificuldades encontradas foram basicamente relacionadas ao entendimento do funcionamento das ferramentas, pois cada programa tem sua abordagem específica para manipulação dos requisitos.

Outro ponto crítico foi a definição das características, sub-características e atributos, que definiram o foco da análise dos softwares selecionados.

7.2 Trabalhos futuros

O usuário selecionado para este trabalho foi a figura do operador, ou seja, a pessoa que de fato opera a ferramenta. Assim, como sugestão para trabalhos futuros, pode-se criar um estudo das ferramentas baseadas em diferentes visões, como a do gerente de desenvolvimento.

Outra sugestão é fazer uma análise a respeito da aplicação de alguma das ferramentas em um ambiente real.

8 Glossário

- Elicitação: descobrir, identificar;
- Java: linguagem de programação;
- Link: ligação;
- Login: efetuar a autenticação para entrar no sistema;
- MySQL: banco de dados;
- Oracle: banco de dados;
- Otimizar: melhorar;
- Pop-up: pequenas janelas que se abrem na tela;
- Prompt: programa destinado à digitação de comandos;
- Templates: formas pré-prontas;

9 Referências bibliográficas

Blaschek, J. R. **Gerência de Requisitos: O principal problema dos projetos de software.** Disponível em: <http://www.bfpug.com.br/islig-rio/Downloads/>. Acesso em 12 de agosto de 2009;

Grande, J. I. de, Martins, L. E., **Sigerar: uma ferramenta para Gerenciamento de Requisitos.** Programa de Mestrado em Ciência da Computação. Disponível em: http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER06. Acesso em: 12 de agosto de 2009;

Kotonya, G., Sommerville I. **Requirements Engineering: Process and Techniques.** Wiley, 1998;

Iam, W., Loomes, M. E. Shankararaman, V., **Managing Requirements Change Using Metrics and Action Planning,** Third European on Software Maintenance, mar. 1999, Amsterdam, Netherlands;

Sommerville, I., Sawyer, P. **Requirements Engineering: A good practice guide.** Wiley, 1999;

Manera, A, F., et al, **Modelo de Qualidade CMMI,** 2007;

Boas, A. V., Gonçalves, J. M., **CMMI para desenvolvimento,** 2006;

Rocha, A. R., Magalhães A. L., **MPS.BR - Melhoria de Processo do Software Brasileiro,** 2007;

Koscianski, A. et al, **Guia para utilização das normas sobre avaliação de qualidade de produto de software – ISO/IEC 9126 e ISO/IEC 14598.** 1999;

Nascimento, A. L., Costa, M. V., Cunha, M. B., **Garantia da Qualidade de Produtos de Software**, s.d.;

Dantas, M. C., **Qualidade e Avaliação de Produto de Software Jurídico**, Trabalho de Graduação, Universidade Federal de Pernambuco, Recife, 2004;

Ribeiro, D. A., **Processo de Avaliação da Qualidade de Arquitetura de Software**, Trabalho de Graduação, Universidade Federal de Pernambuco, Recife, 2005;

Guerra, A. C., Colombo, R. M., **Qualidade de produto de software**;

Filho, M. J., **Um Processo de Avaliação da Qualidade de Arquitetura de Software**, Trabalho de Graduação, Universidade Federal de Pernambuco, Recife, 2005;

Teles, F. S., **Um Processo para Análise de Desempenho de Produtos de Software**, Trabalho de Graduação, Universidade Federal de Pernambuco, Recife, 2005;

Batista, R. M., **Ferramenta de gerência de requisitos de Software integrada com enterprise architect**, Blumenau, 2007;

Lima, A. C., **Análise comparativa de ferramentas livres para adequação às áreas de processo do nível 2 do modelo CMMI**, Passo Fundo, 2009;

Filho, C. V., **Controla: ferramenta de apoio ao processo de desenvolvimento de software em pequenas empresas**, Viçosa, s.d.;

Grings, C. L., Sayão, M., **OpenReq: uma Ferramenta para Auxílio à Gerência de Requisitos**, Rio Grande do Sul, s.d.;

Pereira, S. C., et al, **Um Estudo Empírico sobre Práticas de Engenharia de Requisitos junto a Empresas de Pacotes de Software**, Recife, s.d.;

Castro, J. F., **Gerenciamento de Requisitos**. Disponível em: <http://www.cin.ufpe.br/~if119/aulas>. Acesso em: 25 de setembro de 2009;

Websites:

Jeremia. Disponível em: <http://jeremia.sourceforge.net/>, Acesso em: 18 de setembro de 2009;

OSRMT. Disponível em: <http://sourceforge.net/projects/osrmt/develop>. Acesso em: 19 de setembro de 2009;

SEEC. Disponível em: <http://www.seecforum.unisa.edu.au/SEECTools.html>. Acesso em: 20 de setembro de 2009;

Requirements Management Tools. Disponível em: <http://www.laatuk.com/tools>. Acesso em: 05 de outubro de 2009;

INCOSE. Disponível em: <http://www.incose.org/>. Acesso em: 10 de outubro de 2009;

StickyMinds. Disponível em: <http://www.stickyminds.com>. Acesso em 10 de outubro de 2009;

Ferramentas open source gratuitas. Disponível em: <http://www.mindomo.com>. Acesso em: 15 de outubro de 2009;

Xuse. Disponível em: <http://xuse.sourceforge.net/>. Acesso em: 15 de outubro de 2009;